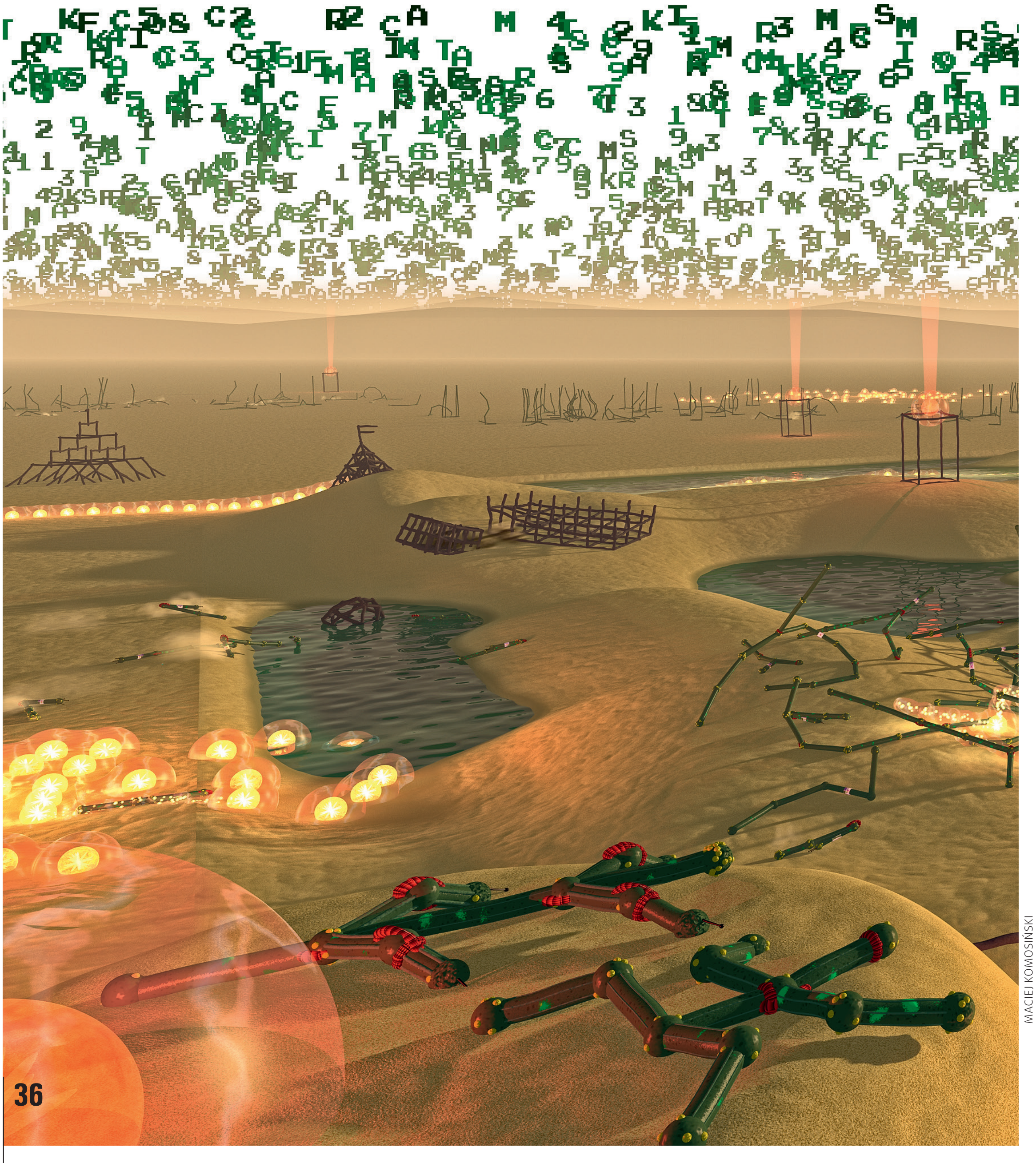
DOI: 10.24425/academiaPAS.2020.135924

Sensitivity in Computer Science

The word "sensitivity" has many meanings, ranging from more mundane technical senses, to meanings specific to statistics and machine learning, all the way to the most human understanding of the concept – that of tender emotions.



MACIEJ KOMOSIŃSKI

Maciej Komosiński

Institute of Computer Science,
Poznań University of Technology

We should start with the caveat that although the word "sensitivity" is certainly used in the field of computer science, it is usually in a completely different sense than human concerns. Although there is a certain analogy involved here – the sensitivities of algorithms, methods or equipment are so described by analogy to human sensitivity – the underlying reference here is to our physical responsiveness to simple stimuli, rather than to our tender feelings. This is much like in the field of photography, where we may talk about the sensitivity of film or a camera sensor, or in the field of physics, where measurement devices may have different sensitivities, different capacities to detect and record signals of a given value (the higher the sensitivity, the weaker or less distinct the signal picked up by the detector).

However, at the end of this essay I will also give an example of sensitivity in computer science which may be surprisingly close to the human kind of sensitivity, involving emotions. But first, let's begin with the standard meaning of "sensitivity" in science and technology.

"Technical" sensitivity

In the technical sense, the sensitivity of a system is understood as the degree to which input values affect output values. This definition applies to a wide variety of devices and mechanisms, such as photographic film (input: light – output: chemical reaction), sensors of digital cameras (light – electrical voltage), algorithms (input data – output data), and also robots (sensor signals – control of effectors, in other words robot behavior).

An algorithm is a procedure for transforming input data into output data. We could list countless examples of specific algorithms that are more sensitive to input conditions, or less so. Moreover, sensitivity may be understood in terms of changes not just in the output data, but also in terms of the very behavior of the algorithm itself, such as the amount of time or memory that the algorithm requires to operate. For example, different algorithms that sort numbers may be sensitive to varying degrees to what series of numbers they receive as input to sort. One sorting algorithm might always take a similar amount of time and utilize a similar amount of memory irrespective of what sequence of numbers of a fixed length it is actually given to sort. Such an algorithm is described as relatively unsensitive to input data – we might say it "doesn't really care" what the input is. Another algorithm may require more time and memory to sort disorderly data, but if we ask it to sort numbers that are already nearly in order, it will do so faster and use less memory. We describe such an algorithm as more sensitive to the input data (see Table 1). Note that the ultimate results of both these algorithms are identical – what we demand from a typical sorting algorithm, after all, is that its outcome should always be ideally sorted – and so in this specific case the difference is in how they do their job, not in the outcome of their work.

Table 1.
An illustration of how an algorithm may be variously sensitive to input data

List to be sorted	Sorting time taken by "sensitive" algorithm	Sorting time taken by "insensitive" algorithm
[1, 3, 2, 4, 5, 6, 7]	3 μs	17 μs
[3, 7, 6, 1, 2, 5, 4]	14 μs	17 μs

One special case of sensitivity, understood as a change in output in response to changed input, is involved in optimization analysis. First we ask what values of the parameters of a certain process will enable us to attain the greatest gain or the lowest cost. Once we know that, we next ask within what bounds the nature of our original problem might vary whilst still ensuring that the solution we discovered remains optimal. By so doing, we check how sensitive this optimal solution is to alterations in the problem as originally posed. As a result, we will know by what magnitude the initial conditions (e.g. costs of production, prices of necessary subcomponents, salaries, staff numbers) can change before this starts to jeopardize the assumed benefits resulting from our intended solution.

As another practical example, we can consider password-verifying algorithms. These, in turn, are algorithms we would like to be sensitive only to a correctly given password, and not sensitive at all to a password that is very nearly correct (e.g. just one letter off from the correct one). Why do we expect insensitivity here? Because if such an algorithm, through its behavior, revealed how close a given attempt is to the real password, hackers making random attempts and observing the behavior of the verification algorithm could deduce clues about which ones are closer to the

Maciej Komosiński, PhD, DSc
works at the Institute of Computer Science at the Poznań University of Technology with a focus on artificial intelligence, building artificial life models, computer simulations, evolution and game theory. The software he has written includes the Framsticks environment and the Top Sailor sailing simulator.
maciej.komosinski@cs.put.poznan.pl

mark, and so they would be able to crack the correct password more quickly. This is a well-known code-breaking technique that has been successfully used in the past.

Additionally, instead of assessing sensitivity by specific, deterministic examples, we may do so in terms of probabilities – as a range of uncertainty in the behavior of the algorithm (more generally system or model) which propagates from input to output. Let's imagine an algorithm predicting tomorrow's weather: the temperature, cloud cover, likelihood of rain. As input, it takes historical data plus current readings from a range of meteorological sensors. We know that the data isn't perfectly precise, so we ask: to what extent does the uncertainty of our forecast depend on the degree of uncertainty of the input data? How inaccurate will our forecast be if the uncertainty of sensors is 5%? How robust and stable is our weather model? These questions can be answered by an analysis of sensitivity to uncertainty. Note that the probabilistic interpretation of sensitivity is very flexible: instead of forecasting weather on the basis of sensor readings, imagine predicting (uncertain) profits made by a company in the following quarter on the basis of (imprecisely reported) sales, or predicting (uncertain) results achieved by an athlete on the basis of (inexact) tests of their ability.

1

Fig. 1
A tiny excerpt from a collection of example images used to teach a machine learning algorithm to recognize different types of fruit

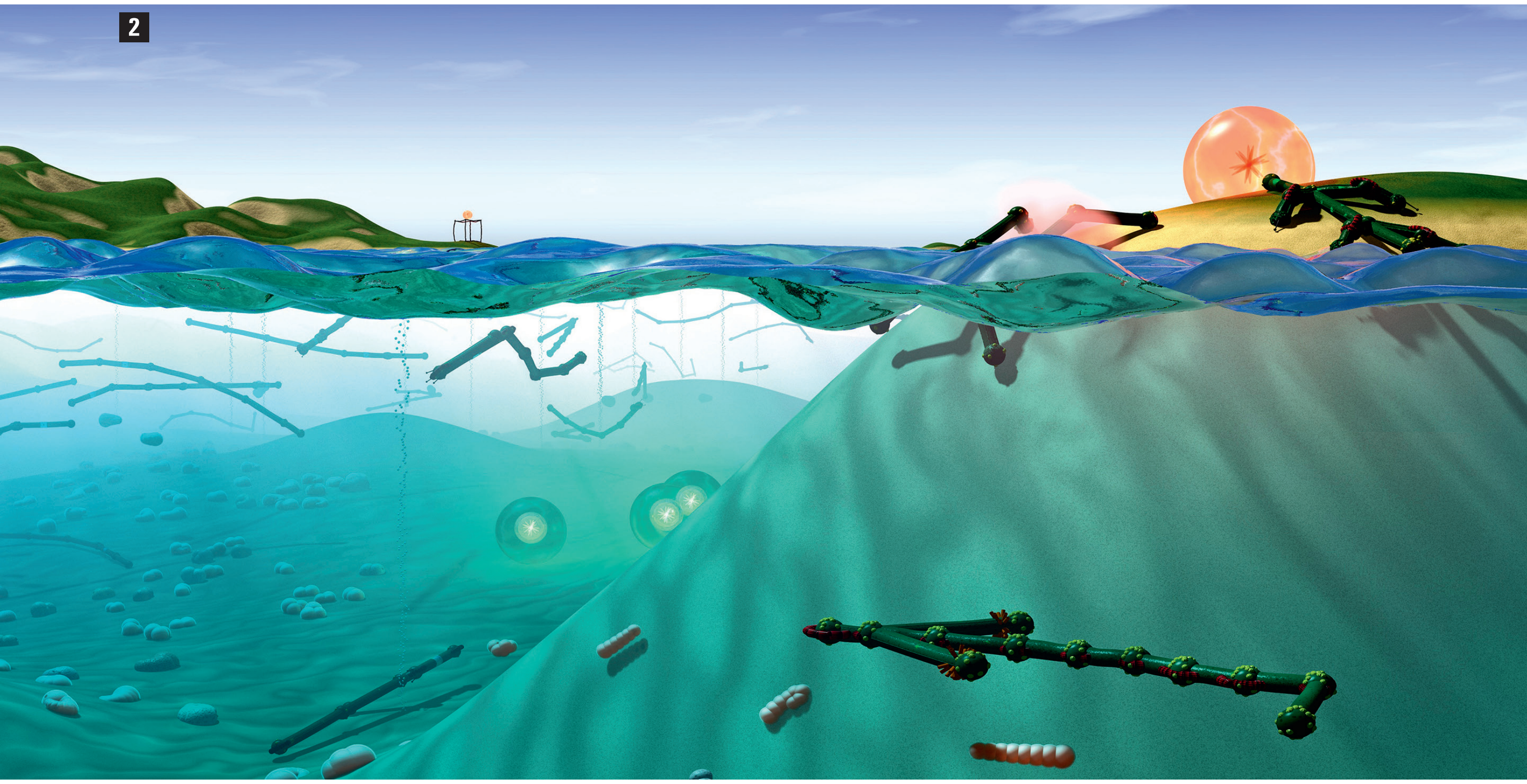
"Statistical" sensitivity

In statistics, sensitivity and specificity are two basic features allowing us to assess the quality of a test. Let's imagine we are testing someone for Lyme disease. We can check for the presence of antibodies in their blood, but does it work? Let's say it does. Does this mean we have a perfect test? It may seem that way, but there is a certain pitfall here: if the test detects antibodies in 100% of positive cases, it means that if they are present in blood, we will always detect them. But we haven't said anything about a situation when the antibodies are not present in blood. If our test were to always categorically state that the antibodies are present, it would certainly always detect them when they are actually there, but it would be a terrible test, since it would have low specificity. Sensitivity doesn't just take account of how well a test detects one of many possible situations (in this instance, actual presence of antibodies), but also how frequently it considers this actual situation to be something else (in this instance, actual presence of antibodies seen as their absence). In other words, the sensitivity of the test tells us how frequently it detects true positives (therefore is correct) in comparison with situations when the result is a false negative. In this case, we want the "sensitivity" to be as high as possible – we want few situations when the test doesn't detect antibodies even though they are there.

The word "sensitivity" is similarly used to assess machine learning algorithms, which mainly learn from data sets (Fig. 1). This may be identifying cancer on the basis of histopathology images, identifying pedestrians on the basis of photos taken by car cameras, identifying bird species from recordings of their songs, defining the concept of happiness using photos of human faces, predicting bankruptcy using financial data, assessing levels of tiredness by how drivers respond to road situations, diagnosing Alzheimer's disease on the basis of the number and variety of words used by patients, and so on. In each of these situations, one of the measures of the algorithm's quality is its sensitivity, which we want to be as high as possible. Low sensitivity has real negative consequences, all of which are easy to imagine for the scenarios listed above.

Sensitivity in complex simulated environments

I promised above that I would talk about an example of sensitivity in computer science which is far closer to our human emotional sensitivity than the technical and statistical senses. To discuss it, we must venture beyond simple computer systems. First we must imagine that researchers manage to understand the



workings of the human brain well enough to be able to produce a faithful simulation. Would such a simulated brain also be able to accommodate tender human sensitivity? If we verify that what we are able to simulate in such a way is all that is necessary to support the latter, then the answer will have to be yes. The only difference will lie in the medium in which the brain operates; its responses and processes will be the same or analogous, and it would be possible to map (model) biochemical phenomena onto phenomena in the brain-simulation medium without any significant loss of information.

Next, let us consider whether such tender sensitivity is a phenomenon limited to just the human species. I think answers can be found in other articles in this issue of *Academia* magazine. If the answer is no, it should be all the simpler to simulate such sensitivity in the case of other species. Although it might be tempting to link sensitivity to consciousness, I will refrain from doing so in this short article (and instead refer readers to a synthetic article[1] I co-authored which provides a clear, formal explanation of various theories of consciousness and discusses its existence in simulations and in computational models of artificial life).

Finally, let's imagine a complex computer-simulated environment (Fig. 2) governed by certain rules we can describe as physical; while they may resemble those found in our own world, they may also be completely different. Such an environment may support evolutionary processes, which may lead to the spontaneous development of simulated "creatures" with increasing complexity.[2] It is possible that such creatures may go on to develop ways of communicating through signals only they understand, and even language. They may also develop emotions, feelings and behaviors common to those creatures, with tender sensitivity possibly figuring among them. But what would this be? After all, the notion of tender sensitivity is defined as a set of behaviors which we humans have ascribed to it. Similarly, a manifestation of sensitivity could arise in our simulated environment, although it may not be the same as ours. If it were to arise, the simulated organisms would know what it is. We, in turn, could agree that their definition of sensitivity is equivalent to our own, if we find an appropriate analogy – this is, after all, precisely what we do every time we talk about sensitivity in animals. ■

Fig. 2
A visualization from Framsticks: a complex simulated environment supporting the development of artificial life

[1] Błądek I., Komosiński M., Miazga K., Mappism: formalizing classical and artificial life views on mind and consciousness, *Foundations of Computing and Decision Sciences* 2019, vol. 44 (1), p. 55–99, http://www.framsticks.com/files/common/MappismConsciousness.pdf

[2] Komosiński M., *Nesting*, 2016, http://www.framsticks.com/nesting